A Point-Based Simulation Framework for Minimally Invasive Surgery

Bo Zhu[1], Lixu Gu[2], Xiaopeng Peng[1], and Zhe Zhou[1]

[1] School of Software, Shanghai Jiao Tong University, China
[2] Med-X Research Institute, Shanghai Jiao Tong University, China
{boolzhu,gulixu,season129,ziziamy}@sjtu.edu.cn

Abstract. In this paper, we present a physically-based simulation framework to build up real-time Minimally invasive surgery (MIS) simulators using point-based techniques. In our framework, simulations of organ contacts, collisions, deformations and haptic feedbacks are all based on discrete physics points. We propose a new smoothed particle hydrodynamics (SPH) approach to simulate non-linear biological soft tissues with a specified tensor computation scheme and experimentally measured organ parameters. We employ a uniform grid method to handle collisions and contacts between organs and surgical instruments represented by particles in real-time. Additionally, a point-based smoothing method for contact feedback is proposed. Our framework could simulate complex surgical scenes in MIS simulation in a unified way, and improve the time efficiency of the entire system without loss of physics accuracy, as shown in our experiment results.

Keywords: Minimally invasive surgery simulation, deformable modeling, point-based simulation, SPH, physically-based modeling.

1 Introduction

Computer-based training system for Minimally invasive surgery (MIS) plays a more and more important part in today's surgical residents' education. And the realism of the virtual surgery scenarios significantly affects the effects of training. In common surgical simulators, different parts of simulation are usually based on different physics or computation models (e.g. FEM [5], BEM [1] or mass-spring [136]) and different data representations (e.g. volume mesh, surface mesh or particle systems). Because there exists hardly any direct correlation between these models, the interaction between them becomes a bottleneck of simulation. To couple these heterogeneous models in one system and let them work together with high performance is a challenging task in today's real-time surgical simulators.

Considering all these drawbacks of model diversity in surgical simulation, in this paper we present a unified framework to represent models and simulate physics behaviors in surgical environment. Our framework is mainly based on point-based simulation techniques, which draws a lot of attention in the field of computation physics,

F. Bello and S. Cotin (Eds.): ISBMS 2010, LNCS 5958, pp. 130–138, 2010.

computer graphics over the past years ([2], [3], [7], [10]). Since most existing point-based applications are focusing on video games, computer animation and off-line physics simulation, there are few research works ([11], [12]) concerning with applying these techniques in the field of real-time biomedical simulation. In our work, a physically-based simulation framework purely using discrete physics points instead of surface or volume meshes is applied in simulating laparoscopic surgery procedures, including simulating organ deformation, controlling instrument movement, detecting collision and computing haptic feedback.

Comparing with the point-based modeling methods used in computer graphics and video games, whose mainly concerns are time efficiency, our modeling approach focus on simulating human organ deformations realistically in virtual surgical environment. To achieve this goal, a new strain-stress tensor computation scheme is combined with SPH method, and material properties of different organs are determined via biomechanics experiments. Our framework contains four main parts: medical data segmentation and point extraction, point-based deformable modeling, collision and haptic handling and visualization feedback. We will discuss each of them in the following.

2 Medical Data Segmentation and Point Extraction

In this stage, organ models necessary for constructing virtual surgery scenes are segmented from the CT image data, and then discrete physics points are extracted from the set of pixels of each segmented organ by using an adaptive sampling algorithm. Different number of sampled points could meet the efficiency requirements in different detail-levels (as in Fig.1). Additionally, surface meshes of organs for visualization are reconstructed by employing marching cube method. In simulation steps, visualization mesh and physics points are combined together by a simple interpolation method (in section 4).

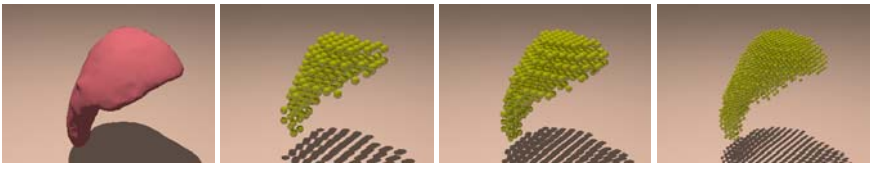

Fig. 1. Surface mesh (first figure on the left) and physics points with different detail levels (three figures on the right) of a liver model are extracted from raw medical data

3 Point-Based Deformable Modeling

With the extracted physics points in the previous step, we use Smoothed Particle Hydrodynamics (SPH), a mesh-free particle system approach to simulate the elasto-mechanical properties of soft bodies in a physically motivated way. Recently similar works in computer graphics could be seen in [3] [2] [7].

3.1 SPH-Based Elasticity Model

SPH method is an interpolation method to approximate properties carried by discrete particles. In SPH method, a deformable object is replaced by a set of particles, and each of them carries a series of physics properties. The local physics property $f(\mathbf{x}_i)$ and its gradient $\nabla f(\mathbf{x}_i)$ of particle i could be approximated by interpolating the properties of its neighbors using the following function:

$$f\left(\mathbf{x}_i\right)=\sum_j \frac{m_j}{\rho_j} f\left(\mathbf{x}_j\right) W\left(\mathbf{x}_i-\mathbf{x}_j, h\right) \tag{1}$$

$$\nabla f\left(\mathbf{x}_i\right)=\sum_j \frac{m_j}{\rho_j} f\left(\mathbf{x}_j\right) \nabla W\left(\mathbf{x}_i-\mathbf{x}_j, h\right) \tag{2}$$

in which $W(\mathbf{x}_i-\mathbf{x}_j, h)$ is kernel function and h is influence radius. In our approach, we use the spiky kernel function as in [3]. The neighbor information and density of each particle is pre-computed before simulation loops.

The way we use to compute the organ deformation based on SPH is similar to [2], in which deformable model is assumed as a Hookean material. The strain ε and the stess σ follow the following rules:

$$\sigma=\mathbf{C}\varepsilon \tag{3}$$

in which **C** is a rank four tensor. As for isotropic materials, **C** could be represented by a 6x6 matrix determined by Young's Modulus and Poisson Ratio of linear elastic materials as in [2] and [7]. In our case, we use coefficients λ_1, λ_2 and λ_3 determined from non-linear anatomical tissue experiments (in Part 3.2) to compute the tensor C.

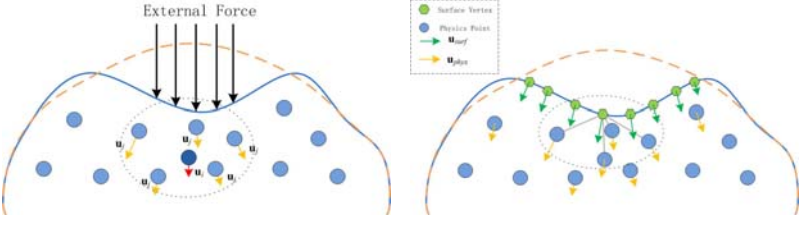

Fig. 2. The left figure shows a solid (blue lines) deforms from the rest shape (orange line) due to the external force. The displacement of each physics point is computed according to the displacements of its neighbors within distance h. The right figure illustrates the displacements of surface vertices (green) are computed based on the displacements of the neighbor physics points (blue) as in section 5.

SPH method is employed to approximate the derivatives of displacement of each particle:

$$\nabla \mathbf{u}_i=\sum_j \frac{m_j}{\rho_j}\left(\mathbf{u}_j-\mathbf{u}_i\right) \nabla W\left(\mathbf{x}_i^0-\mathbf{x}_j^0, h\right)^T \tag{4}$$

where ρ is the density of particle *i*, which is pre-computed using the information of the referenced rest shape of the object.

With $\nabla \mathbf{u}_i$, ε_i is computed as Green-Saint-Venant strain tensor with the form

$$\varepsilon = \nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u} \nabla \mathbf{u}^T \tag{5}$$

Stress tensor σ_i is calculated using equation (3). With both strain and stress, we could get the strain energy density on each point and the elastic force is computed as the derivative of the energy density with respect to the direction of displacement **u** as in [2]. In detail, the elastic force $\mathbf{f}_{ji}$ exerted on particle j by particle *i* is computed as

$$\mathbf{f}_{ji} = -\nabla \mathbf{u}_i U_i = -\frac{m_i}{\rho_i}(\mathbf{I} + \nabla \mathbf{u}_i)\sigma_i \mathbf{d}_{ij} \tag{6}$$

With

$$\mathbf{d}_{ij} = \frac{m_j}{\rho_j} \nabla \left(\mathbf{x}_i^0 - \mathbf{x}_j^0, h \right) \tag{7}$$

In each time step, the forces including elastic force, body force and external force on each particle are summed up and applied in a common ODE solver to evolve the system with time (in Fig. 3). To improve the numerical stability, we employ XSPH [4] to smooth the interpolation values and use a second order Runge-Kutta integration scheme in the ODE solver.

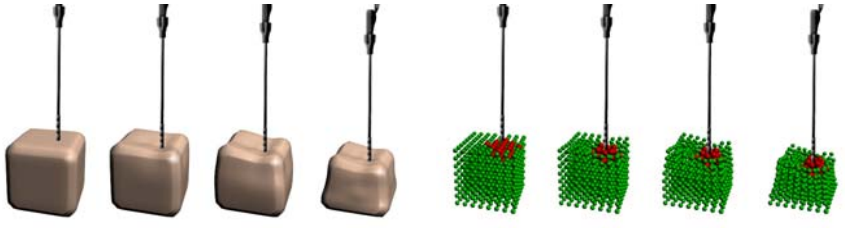

Fig. 3. An elastic cube deforms under pressure of a nipper. Our point-based elastic model and the experimental material parameters are employed. Images are taken every 20 timesteps with (left 4 pictures) and without (right 4 pictures) combined surface rendering.

3.2 Physically Accurate Organ Elastic Tensor

In most applications simulating soft body in computer graphics, the material properties are simply controlled by the Poisson Ratio and Young's Modulus embedded in tensor **C**. However, as for non-linear materials such as biological soft tissues, this is not accurate enough. In our method, we compute the tensor **C** based on polynomial energy functions with parameters determined by biological mechanics experiments. The tensor **C** is computed using the following equation

$$\mathbf{C} = \begin{bmatrix} \eta+2\mu & \eta & \eta & & & & \\ \eta & \eta+2\mu & \eta & & & & \\ \eta & \eta & \eta+2\mu & & & & \\ & & & \mu & & & \\ & & & & \mu & & \\ & & & & & \mu & \end{bmatrix} \tag{8}$$

where

$$\mu = -2\lambda_2 \tag{9}$$

$$\eta = 4\lambda_1\lambda_3{}^2 e^{2\lambda_3(\varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33})} + 4\lambda_2 \tag{10}$$

We determine the coefficients λ_1, λ_2 and λ_3 for different anatomical organs by measuring the displacement-load relations of tissues sampled from different parts of porcine kidney and liver on INSTRON mechanics equipment and fitting the strain-stress curve using polynomial interpolation method. We get the parameters for kidney and liver model used in our surgical environment as in Table 1.

Table 1. Empirical parameters for different organ models

	λ_1	λ_2	λ_3
Liver	1788.32	-1981.95	6.712
kidney	1646.02	3503.93	5.569

With the measured parameters to represent tensor C, it is possible to describe the specific deformation behaviors of specific organs and tissues.

4 Collision Handling, Haptic Feedback and Visualization

Detecting collision and simulating the interactions between organs and surgical instruments is a very important part in real-time surgical simulation. In our point-based framework, all the objects are represented as points carrying physical properties, so we use a uniform grid to compute their interactions.

The uniform grid (as in Fig.4) subdivides the space of the laparoscopic surgical environment into N^3 voxels with the same size. N is computed as N = L / h, in which L is the longest length of space and h is the influence radius used in SPH. In each time step, for each particle *i*, the indexes x, y, z of voxel **b**(x, y, z) where it locates is updated and the collisions of particle *i* is detected and handled locally within the 27 neighbor voxels. Uniform grid method requires more memories, but it is a highly parallelizable method which could be easily implemented on GPGPU.

After detecting collisions in uniform grid, a correct response should be made both visually and haptically. Since our collision response method is impulse-based as in [9], drastic changes of response force usually occur when the time step is large, which

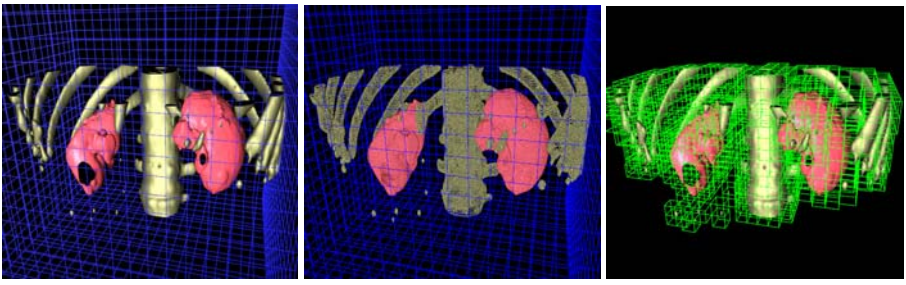

Fig. 4. The surgical environment is modeled as a uniform grid (blue grid in left and middle picture). The model is visualized as a surface mesh (left), physics points are inserted in to the grid (middle), and the voxel containing physics points (green) are updated in each time step (right).

will make the haptic feedback discontinues. On the other hand, in point-based methods, external force is usually added on one discrete physics point, which will lead to various inconsistencies because the contact area is a finite area rather than a infinite point in real world. To overcome these drawbacks, we integrate pressure mask, a method used in boundary element method [1] to smooth the force distribution on the surface, into our framework to get a smoothed haptic feedback from the deformed anatomical organ.

The basic idea of pressure mask is very simple: a smoothed scalar function is used to distribute a vector on a single discrete point to a continuous local area. The scalar function should satisfy the normalizing condition and smoothness condition with compact support. In our framework, we simply reuse the SPH kernel function in deformable solid simulation as the force mask. The external force exerted on each neighbor particle is calculated as

$$\mathbf{f}_j = \frac{W\left(\mathbf{x}_i - \mathbf{x}_j, h\right)}{\sum_j W\left(\mathbf{x}_i - \mathbf{x}_j, h\right)}\mathbf{f} \tag{11}$$

After mask the external force in a local area, the force exerted on one single point is distributed smoothly on a set of neighbor points. And we can get the continuous haptic feedback over time steps (illustrated in Fig. 6).

Visualization in our framework is implemented by combining physics points and renderable surface mesh in real-time. The combine method is based on interpolation between vertexes of surface meshes and physics points. The SPH kernel function is used to sum up the weighted displacements of neighbor physics points and this sum is regarded as the displacement of the surface vertex:

$$\mathbf{u}_i^{surf} = \frac{\sum_j W\left(\mathbf{x}_i - \mathbf{x}_j, h\right)\mathbf{u}_j^{phyx}}{\sum_j W\left(\mathbf{x}_i - \mathbf{x}_j, h\right)} \tag{12}$$

in which $\mathbf{u}_i^{surf}$ is the displacement of surface vertex and $\mathbf{u}_j^{phyx}$ is the displacement of its neighbor physics points (illustrated in Fig. 2). This method is very fast and easy to implement. And high resolution visual feedback of surgical scenes (in Fig. 5) could be provided from a small number of physics points. But this method is weak in preserving the deformation details of the surface. To improve this condition, adaptive particle methods [13] could be applied.

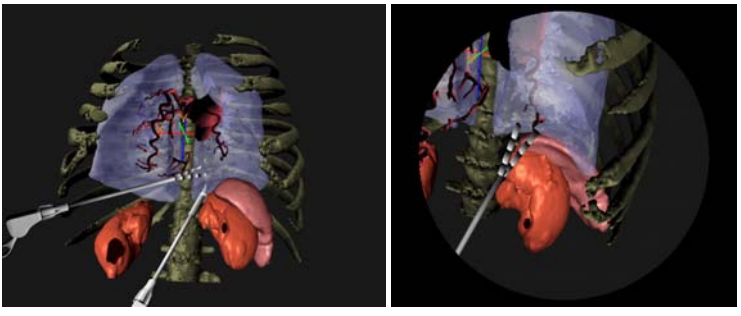

Fig. 5. A Minimally invasive surgery simulator based on our point-based simulation framework

5 Experiments and Results

As the physics accuracy of visualization and haptic feedback are demonstrated in section 3 and 4, in this part we will focus on the scalability and time performance of our framework. All the experiments were carried out on a 2.26GHz Pentium M notebook PC with GeForce 9650M graphics card and 2 GB of memory.

First, we test the scalability of our deformable model by simulating models with different number of physics points. We generate deformable cubes (Fig. **3**) with physics points increasing linearly from 500 to 3000. Then an external force is added on its surface and the time cost of processing deformation is recorded in each time step. The process includes deformation computing, updating surface meshes and generating

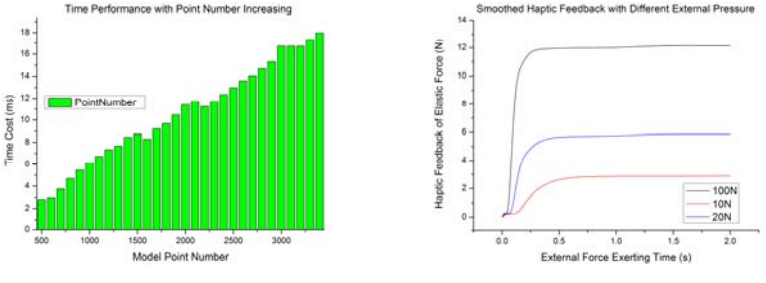

Fig. 6. The left figure illustrates time cost of deformation computing increases linearly with the number of physics points. The right figure shows smoothed elastic response force of the contact point after employing the pressure mask method as in section 4.

haptic feedback. Space grid updating and rendering time are not taken into account. As illustrated in Fig. 6, the time cost meets the real-time requirements well, and increases linearly with the number of points increasing. Comparing with the $O(n^2)$ time complexity of FEM or BEM model, this demonstrates that the speedup of the point-based model in our framework is noticeable even for large number of models.

To evaluate the time performance of different parts of our framework, we extend the previous test to an integrated MIS environment including interaction and rendering (Fig. 5). Then we record the time cost of different parts in each time step as in Table 2.

Table 2. Time performance of different parts in an integrated simulation

Physics-Num	Deformation (ms)	Collision (ms)	Integration (ms)	Rendering (ms)
1000	5.3099	0.7799	0.1422	0.4700
2000	11.3999	1.8800	0.1452	0.4599
3000	17.6499	2.8099	0.1503	0.4700
4000	23.7500	4.0699	0.1540	0.4700
5000	30.1599	5.0000	0.1600	0.6200

The results show that the main computation resource is focusing on physically-based computing, while the time cost of collision handling (including updating uniform grid), integrating and rendering part is tiny comparing with the deformation part. Something worth notice is that the time cost of integration is nearly constant with the same surface mesh. This means that it is easy to couple a fine sampled particle system with a surface mesh to enhance the realism of deformation without increasing the integrating cost. The MIS simulator with our framework could work at 60 fps in average, and this proves to be suitable for real-time applications.

6 Conclusion

In this paper we propose a point-based simulation framework for MIS simulators. We integrate point-based techniques in model construction, deformable simulation, collision handling and haptic rendering into a unified framework to simulate the physics behaviors in MIS system. In this framework, our current work is focusing on simulating organ deformations and handling interactions. Future work will be related to integrating blood simulation into this framework to enhance reality and use GPGPU to improve the time efficiency because most of the point-based algorithms in our framework are parallelizable.

References

1. James, D.L., Pai, D.K.: ArtDefo: accurate real time deformable objects. In: SIGGRAPH 1999: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 65–72. ACM Press/Addison-Wesley Publishing Co., New York (1999)

2. Becker, M., Ihmsen, M., Teschner, M.: Corotated SPH for Deformable Solids. In: Proc. Eurographics Workshop on Natural Phenomena (2009)
3. Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M., Alexa, M.: Point based animation of elastic, plastic and melting objects. In: SCA 2004: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 141–151. Eurographics Association, Aire-la-Ville (2004)
4. Monaghan, J.: On the Problem of Penetration in Particle Methods. JCP 82, 1–15 (1989)
5. Nesme, M., Marchal, M., Promayon, E., Chabanas, M., Payan, Y., Faure, F.: Physically realistic interactive simulation for biological soft tissues (2005)
6. Pang, W.-M., Qin, J., Chui, Y.-P., Wong, T.-T., Leung, K.-S., Heng, P.-A.: Orthopedics surgery trainer with PPU-accelerated blood and tissue simulation. In: Int. Conf. Med. Image Comput. Comput. Assist. Interv., vol. 10(Pt 2), pp. 842–849 (2007)
7. Solenthaler, B., Schlдfli, J., Pajarola, R.: A unified particle model for fluid–solid interactions: Research Articles. Comput. Animat. Virtual Worlds 18(1), 69–82 (2007)
8. Turini, G., Pietroni, N., Megali, G., Ganovelli, F., Pietrabissa, A., Mosca, F.: New techniques for computer-based simulation in surgical training. International Journal of Biomedical Engineering and Technology, IJBET (2008)
9. Choi, K.-S., Sun, H., Heng, P.-A.: An efficient and scalable deformable model for virtual reality-based medical applications. Artificial Intelligence in Medicine 32(1), 51–69 (2004)
10. Harada, T., Tanaka, M., Koshizuka, S., Kawaguchi, Y.: Real-time Coupling of Fluids and Rigid Bodies. In: APCOM 2007 in conjunction with EPMESC XI, Kyoto, Japan, December 3-6 (2007)
11. Horton, A., Wittek, A., Miller, K.: Subject-specific biomechanical simulation of brain indentation using a meshless method. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 541–548. Springer, Heidelberg (2007)
12. Lim, Y.-J., De, S.: Real time simulation of nonlinear soft tissue response in minimally invasive surgical procedures using a meshfree approach. In: Proceedings of the US National Congress on Computational Mechanics, Austin, TX (2005)
13. Adams, B., Pauly, M., Keiser, R., Guibas, L.J.: Adaptively sampled particle fluids. In: SIGGRAPH 2007: ACM SIGGRAPH 2007 papers, p. 48. ACM, New York (2007)